Lincoln School of Computer Science

CMP2801M - Advanced Programming - Assessment Item 1

Learning Outcome	Criterion	Pass	2:2	2:1	1st	1st++
[LO1] Apply concepts of advanced software development and programming methods to computational problems (40%)	Implementation (40%)	The code compiles and executes. Fair program structure and some code comments. A working software with basic functionality is demonstrated, accomplishing the assignment tasks partially.	The code compiles and executes. Clear program structure and appropriate comments. A working application is demonstrated, accomplishing most of the tasks.	The code compiles and executes. The program code is well structured and commented. Good demonstration of C++ features, such as collections, pointers and memory management. The functionality of the code is almost complete.	A complete implementation featuring all the desired functionality. The program code uses appropriate evaluation strategies, is well-structured and commented. Good use of programming structures and C++ features is demonstrated. The stretch task may have been attempted.	The application is a production-worthy solution that fulfils all elements of the brief and makes effective use of STL/contemporary C++ components. An innovative solution to stretch task is also provided.
[LO2] Use advanced object oriented principles and programming techniques in software development	Implementation (40%)	Simple class structures are used. Those classes may incorporate variables and data structures, but no thought has been given to their OOP principles.	Class definitions are appropriate. Inheritance relationships are implemented, although with maybe flawed or missing elements. Access modifiers are in place but may not be correctly used.	Class definitions are appropriate. Inheritance relationships are implemented with minor flaws. Access modifiers are correctly used in the classes. Virtual functions have been used correctly, and one or more advanced techniques have been used (e.g., operator overloading).	Classes are defined to match the assignment brief perfectly. The inheritance relationships are correctly implemented. Access modifiers are correctly and effectively used. Several advanced techniques (e.g., operator overloading) are demonstrated and effectively utilised in the application.	The implementation demonstrates advanced knowledge of both OOP and functional paradigms (i.e. lambdas). Code repetition is kept to an absolute minimum and careful thought has gone into the application's extensibility.
[LO 3] Apply advanced logical and mathematical techniques in the development of software solutions	Report (20%)	Introduction is basic, introducing the problem but little more. Code structure is incomplete. Testing is limited in scope, and almost no attempt is made to evaluate the program.	The introduction provides an overview of the problem. Program structure and logic is described in a satisfactory way. Test results are presented but the analysis of the program is lacking.	The introduction provides an explanation of the problem. Program structure and logic is described well. Test results are presented, and an analysis of the program's efficiency is given.	The introduction provides a detailed explanation of the problem. Program structure and logic is described very well. Test results are presented clearly, and the program is comprehensively evaluated.	The report is written in clear, technical language throughout. A detailed examination of the additional tasks is also given, and the program design is critically evaluated in terms of time/space complexity.
Weighting		Weightings are indicated				

Document updated 1 August 21